CLAIM AMENDMENTS

1.   (Original)   A method of increasing throughput of a server capable of

servicing at least one TCP/IP connection with a client, the server creating a TCP/IP

Transmission Control Block (TCB) stored in non-paged pool (NPP) memory containing

information required to identify and to service the client connection, comprising the steps of:

closing a TCP/IP connection;

excluding information from the TCB not required to identify the client connection to

form a timed-wait state TCB (TWTCB) for a time-wait period; and

releasing the NPP memory containing the information required to service the client

connection.

2.   (Original)   The method of claim 1, wherein the step of excluding comprises

the step of copying the information required to identify the client connection to form the

TWTCB.

3.   (Original)   The method of claim 2, wherein the step of releasing the NPP

memory containing the information required to service the client connection includes the step

of releasing the NPP memory of the TCB required to identify the client connection.

4.   (Original)   The method of claim 1, wherein the step of excluding

information not required to identify the client connection to form a TWTCB comprises the

step of maintaining a minimum of information necessary to avoid late-routed packets

forming new connections on the server.

5.   (Original)   The method of claim 1, wherein the step of excluding information

not required to identify the client connection to form a TWTCB comprises the step of

establishing a TWTCB of the following structure:

```
struct TWTCB {

#ifdef DEBUG
    ulong       twtcb_sig;
#endif
    struct TWTCB    *twtcb_next;
    IPAddr      twtcb_daddr;     // Destination IP address.
    ushort      twtcb_dport;     // Destination port.
    ushort      twtcb_sport;     // Source port.
    uint        twtcb_partition;
    ushort      twtcb_delta;
    ushort      twtcb_rexmittimer;
    Queue       twtcb_TWQueue;  // Place to hold all the timed_waits
    uint        twtcb_flags;
    IPAddr      twtcb_saddr;     // Source IP address.
    SeqNum      twtcb_senduna;
#if 0 // TRIM_TWTCBREMOVE
    SeqNum      twtcb_sendnext;
#else
    struct TWTCB    *twtcb_prev;
#endif
    SeqNum      twtcb_rcvnext;
    uint        twtcb_phxsum;      // Precomputed pseudo-header xsum.
    DEFINE_LOCK_STRUCTURE(twtcb_lock)

    //ulong       twtcb_refcnt;
    //SeqNum      twtcb_sendmax;
    //uchar       twtcb_state;     // State of this TCB.
    //RouteCacheEntry  *twtcb_rce;     // RCE for this connection.
};
```

6.   (Original)   The method of claim 1, wherein the step of excluding information

not required to identify the client connection to form a TWTCB comprises the step of

establishing a TWTCB of the following structure:

```
struct TWTCB {

#ifdef DEBUG
  ulong    twtcb_sig;
#endif
  struct TWTCB  *twtcb_next;
  IPAddr  twtcb_daddr;        // Destination IP address.
  ushort  twtcb_dport;        // Destination port.
  ushort  twtcb_sport;        // Source port.
  ushort  twtcb_delta;
  ushort  twtcb_rexmittimer;
  IPAddr twtcb_saddr;         // Source IP address.

  //ulong  twtcb_refcnt;
  //SeqNum  twtcb_sendmax;
  //uchar  twtcb_state;        // State of this TCB.
  //RouteCacheEntry      *twtcb_rce;      // RCE for this connection.

};
```

7.   (Original)   The method of claim 1, wherein the step of excluding information

not required to identify the client connection comprises the step of forming a TWTCB that

occupies less memory than the TCB.

8.   (Original)   The method of claim 7, wherein the step of forming a TWTCB that occupies less memory than the TCB comprises the step of forming a TWTCB that occupies approximately 96 bytes of memory.

9.   (Original)   The method of claim 7, wherein the step of forming a TWTCB that occupies less memory than the TCB comprises the step of forming a TWTCB that occupies approximately 64 bytes of memory.

10.   (Original)   The method of claim 7, wherein the step of forming a TWTCB that occupies less memory than the TCB comprises the step of forming a TWTCB that occupies approximately a single cache line.

11.   (Original)   A method for increasing the throughput of a server capable of servicing at least one TCP/IP connection, the server establishing a TCP/IP Transmission Control Block (TCB) of a size and containing information sufficient to identify and service the connection, comprising the steps of:

closing the at least one TCP/IP connection;

forming a Timed-Wait TCB (TWTCB) of a size less than the TCB; and

releasing the TCB for use by the server.

12.   (Original)   The method of claim 11, wherein the step of forming a TWTCB comprises the step of copying a portion of the information of the TCB, the portion of information

5

being sufficient to identify the TCP/IP connection to prevent late routed packets from forming

new connections.

13.   (Original)   The method of claim 12, wherein the TCB occupies approximately

440 bytes of memory, and wherein the step of forming a TWTCB comprises the step of forming

a TWTCB that occupies approximately 206 bytes of memory.

14.   (Original)   The method of claim 12, wherein the TCB occupies approximately

440 bytes of memory, and wherein the step of forming a TWTCB comprises the step of forming

a TWTCB that occupies approximately 32 bytes of memory.

15.   (Original)   The method of claim 11, wherein the step of forming a TWTCB

comprises the step of forming a TWTCB having the following structure:

```
struct TWTCB {

#ifdef DEBUG
    ulong        twtcb_sig;
#endif
    struct TWTCB    *twtcb_next;
    IPAddr       twtcb_daddr;      // Destination IP address.
    ushort       twtcb_dport;      // Destination port.
    ushort       twtcb_sport;      // Source port.
    uint         twtcb_partition;
    ushort       twtcb_delta;
    ushort       twtcb_rexmittimer;
    Queue        twtcb_TWQueue;  // Place to hold all the timed_waits
    uint         twtcb_flags;
    IPAddr       twtcb_saddr;       // Source IP address.
    SeqNum       twtcb_senduna;
```

```
#if 0 // TRIM_TWTCBREMOVE
   SeqNum      twtcb_sendnext;
#else
   struct TWTCB   *twtcb_prev;
#endif
   SeqNum       twtcb_rcvnext;
   uint         twtcb_phxsum;      // Precomputed pseudo-header xsum.
   DEFINE_LOCK_STRUCTURE(twtcb_lock)

   //ulong      twtcb_refcnt;
   //SeqNum     twtcb_sendmax;
   //uchar      twtcb_state;    // State of this TCB.
   //RouteCacheEntry  *twtcb_rce;    // RCE for this connection.
};
```

16.   (Original)   The method of claim 11, wherein the step of forming a TWTCB·

comprises the step of forming a TWTCB having the following structure:

```
struct TWTCB {

#ifdef DEBUG
   ulong    twtcb_sig;
#endif
   struct TWTCB  *twtcb_next;
   IPAddr  twtcb_daddr;       // Destination IP address.
   ushort  twtcb_dport;       // Destination port.
   ushort  twtcb_sport;       // Source port.
   ushort  twtcb_delta;
   ushort  twtcb_rexmittimer;
   IPAddr  twtcb_saddr;       // Source IP address.

   //ulong  twtcb_refcnt;
   //SeqNum  twtcb_sendmax;
   //uchar  twtcb_state;       // State of this TCB.
   //RouteCacheEntry    *twtcb_rce;    // RCE for this connection.

};
```

17. (Original) The method of claim 11, wherein the step of forming a TWTCB comprises the step of copying a portion of the information of the TCB, the portion of information being insufficient to service the TCP/IP connection.

18. (Original) A computer readable medium having computer-executable instructions for performing steps, comprising:

closing a TCP/IP connection;

copying less than all information stored in a TCP/IP Transmission Control Block (TCB) into a Timed-Wait TCB (TWTCB); and

maintaining the TWTCB for a timed-wait period to avoid late-routed packets from establishing a new connection with a server.

19. (Original) The computer-readable medium of claim 18, wherein the step of copying less than all the information stored in a TCB into a TWTCB comprises the step of copying information sufficient to uniquely identify the TCP/IP connection.

20. (Original) The computer-readable medium of claim 18, further comprising the step of releasing memory used to store the TCB for use by the server after the step of copying less than all of the information stored in the TCB into a TWTCB.

8

21. (Original) The computer-readable medium of claim 18, wherein the step of

copying less than all the information stored in a TCB into a TWTCB results in a structure for the

TWTCB that fits on one cache line.

22. (Original) A computer-readable medium having stored thereon a data

structure, consisting essentially of:

```
struct TWTCB {

#ifdef DEBUG
    ulong        twtcb_sig;
#endif
    struct TWTCB    *twtcb_next;
    IPAddr        twtcb_daddr;        // Destination IP address.
    ushort        twtcb_dport;        // Destination port.
    ushort        twtcb_sport;        // Source port.
    uint          twtcb_partition;
    ushort        twtcb_delta;
    ushort        twtcb_rexmittimer;
    Queue         twtcb_TWQueue;  // Place to hold all the timed_waits
    uint          twtcb_flags;
    IPAddr        twtcb_saddr;        // Source IP address.
    SeqNum        twtcb_senduna;
#if 0 // TRIM_TWTCBREMOVE
    SeqNum        twtcb_sendnext;
#else
    struct TWTCB    *twtcb_prev;
#endif
    SeqNum        twtcb_rcvnext;
    uint          twtcb_phxsum;        // Precomputed pseudo-header xsum.
    DEFINE_LOCK_STRUCTURE(twtcb_lock)

    //ulong        twtcb_refcnt;
    //SeqNum       twtcb_sendmax;
    //uchar        twtcb_state;      // State of this TCB.
    //RouteCacheEntry  *twtcb_rce;      // RCE for this connection.
};
```

23.   (Original)   A computer-readable medium having stored thereon a data

structure, consisting essentially of:

```
struct TWTCB {

#ifdef DEBUG
  ulong    twtcb_sig;
#endif
  struct TWTCB  *twtcb_next;
   IPAddr  twtcb_daddr;        // Destination IP address.
   ushort  twtcb_dport;        // Destination port.
   ushort  twtcb_sport;        // Source port.
   ushort  twtcb_delta;
   ushort  twtcb_rexmittimer;
   IPAddr  twtcb_saddr;        // Source IP address.

   //ulong   twtcb_refcnt;
   //SeqNum  twtcb_sendmax;
   //uchar   twtcb_state;       // State of this TCB.
   //RouteCacheEntry     *twtcb_rce;       // RCE for this connection.

};
```